**Task 1**

**1. based on the scenario provided.**

an algorithm is

A sequence of steps used to solve a problem is called an algorithm. In this case, I'm using a "for" loop to retrieve the number of lines in the "summary.py" file as well as inputs from text files line by line. After that, the system will deliver the anticipated outcomes using simple mathematics. An algorithm is a mathematical procedure used in this circumstance to solve a problem.

**II. The procedure for creating an application**

1. Recognized the query

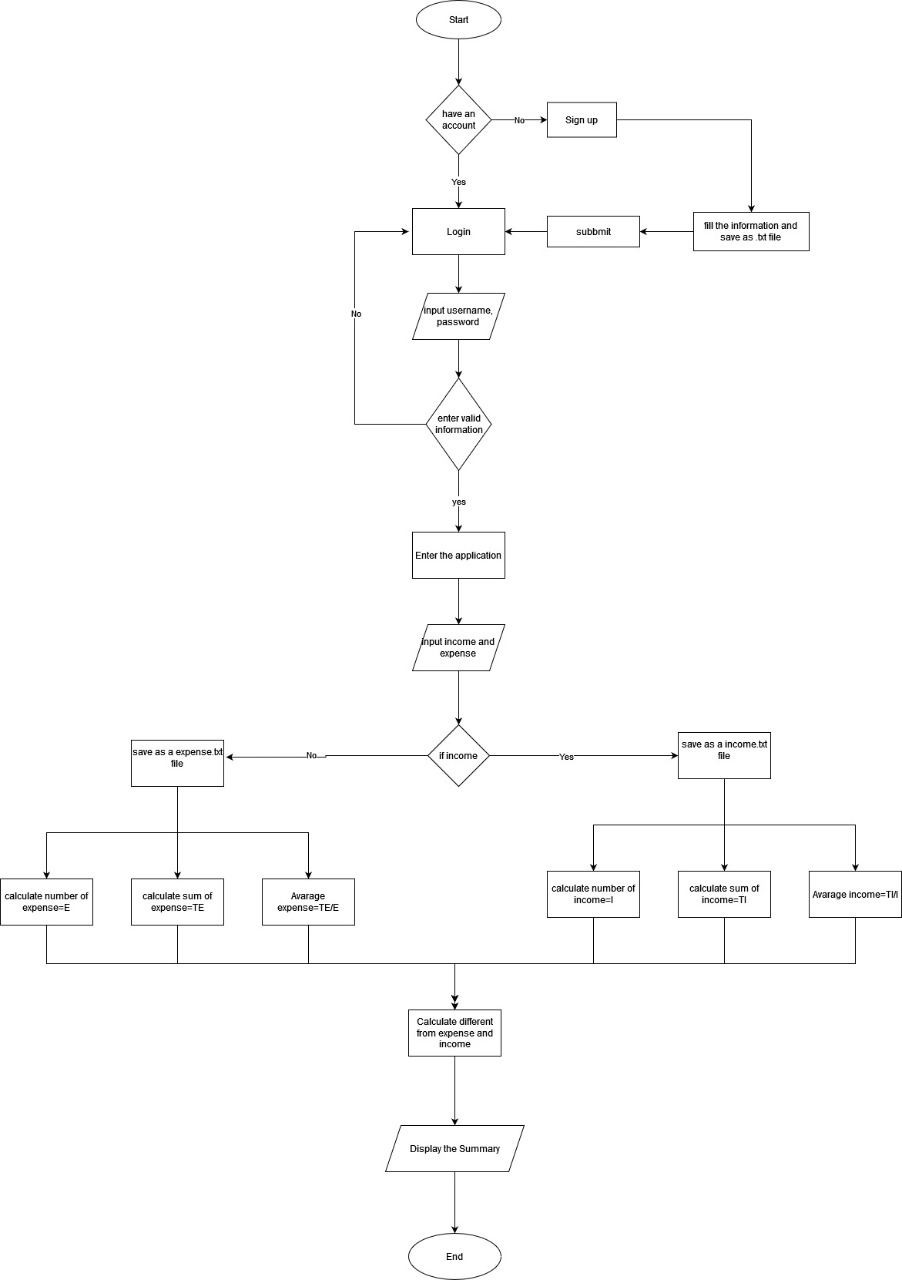
2. gathered specifications

3. Requirements analysis 4. Algorithm development

5. Flowcharted the system representation

6. Python was used to create the system.

**3.write the algorithm to reflect the above scenario using pseudo code or flow chart**



**Analyze the relationship between the written algorithm and the code variation as well as the implementation of the algorithm in a suitable language.**

**When referring to code in computer science, the term "algorithm" is typically used to describe a highly optimized design, such as a widely utilized solution to a challenging problem. The phrase describes software that adheres to stringent design guidelines and solves challenging issues.**

**Code is typically repetitive and basic. Codes, for instance, are frequently easy to apply. Algorithms are more complex and may start as a research project or another important undertaking. An algorithm is typically not considered to be any impromptu code produced by a developer that does not solve a significant problem.**

**Task 2**

**2.1 Describe the procedural, object-oriented, and event-driven paradigms, as well as their similarities and differences.**

**In procedural programming, a program is defined as a collection of phases, including its subprograms. Conversely, declarative programs aim to describe the output without regard for the procedures required to computerize it, instead providing some description or denotation of the desired outcome.**

**A technique for organizing code based on encapsulation, inheritance, replacement, programming to interfaces, and other concepts is known as object-oriented programming. Most procedural programming is used in object-oriented programming.**

**Event-based programming requires creating event handling procedures and utilizing the core event loop of the underlying system. By utilizing numerous libraries, you can avoid the challenge of creating your own event loop. They currently work with the event loop given by the system. Even if it is not always the case, object-oriented programming is regularly used to create event-based applications.**

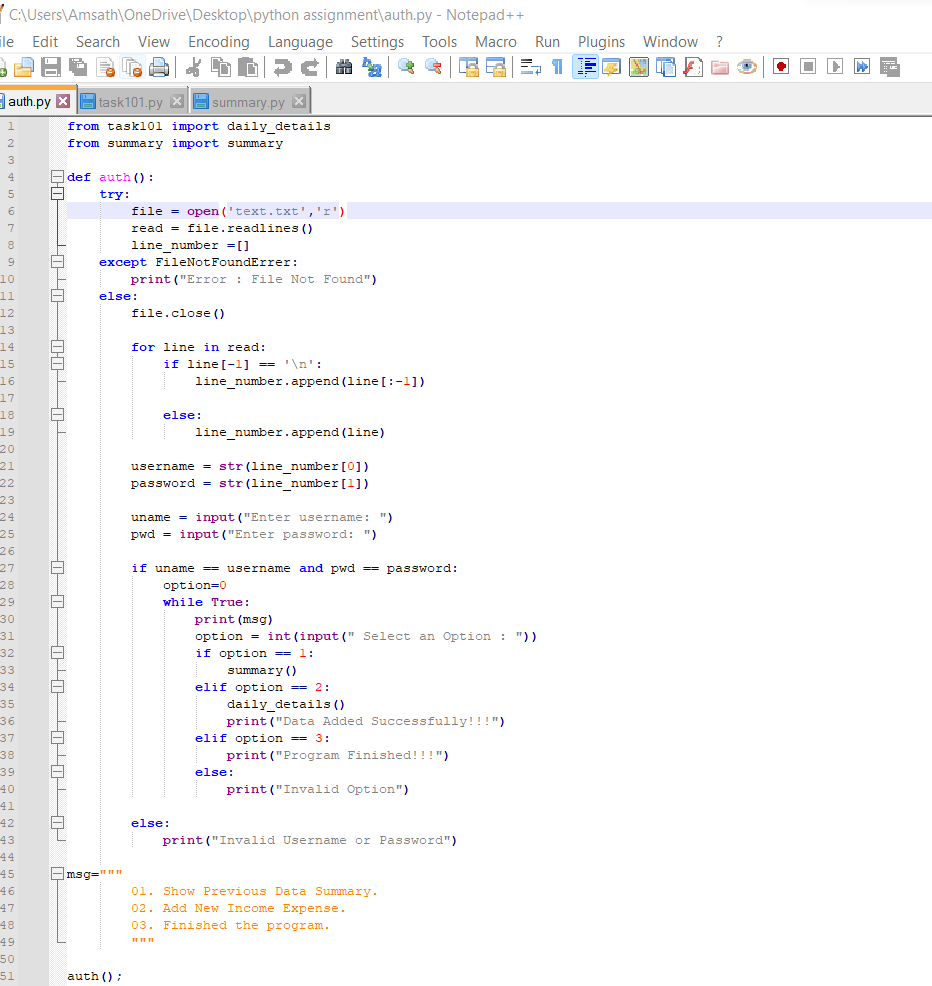
**2.2 Based on your Task 1 algorithms, compare and contrast the procedural, object-oriented, and event-driven paradigms.**

**Small functions make up the program, and the code is reusable. Data abstraction is a notion used by this system. Simple to add features. The system is very nearly object-oriented.**

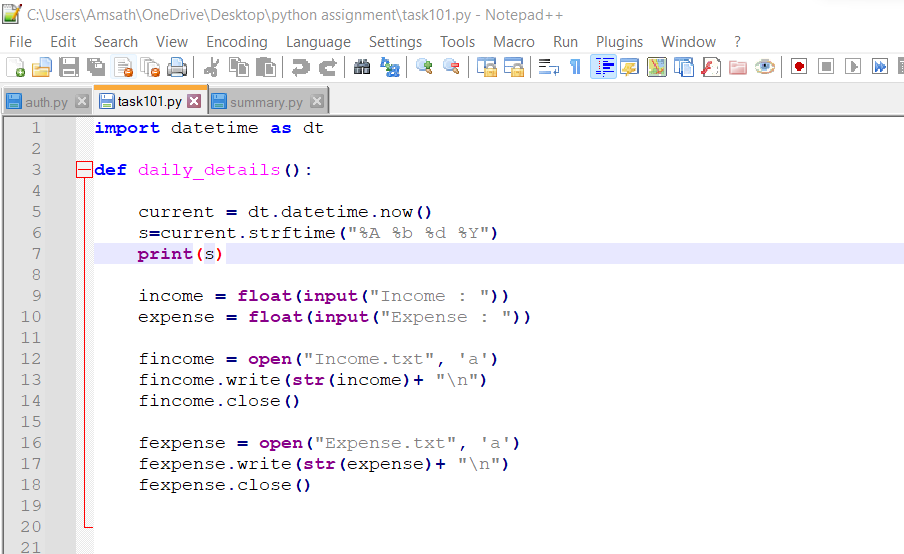
**Task 3**

**Implement your solution as per the requirements specified in the scenario.**

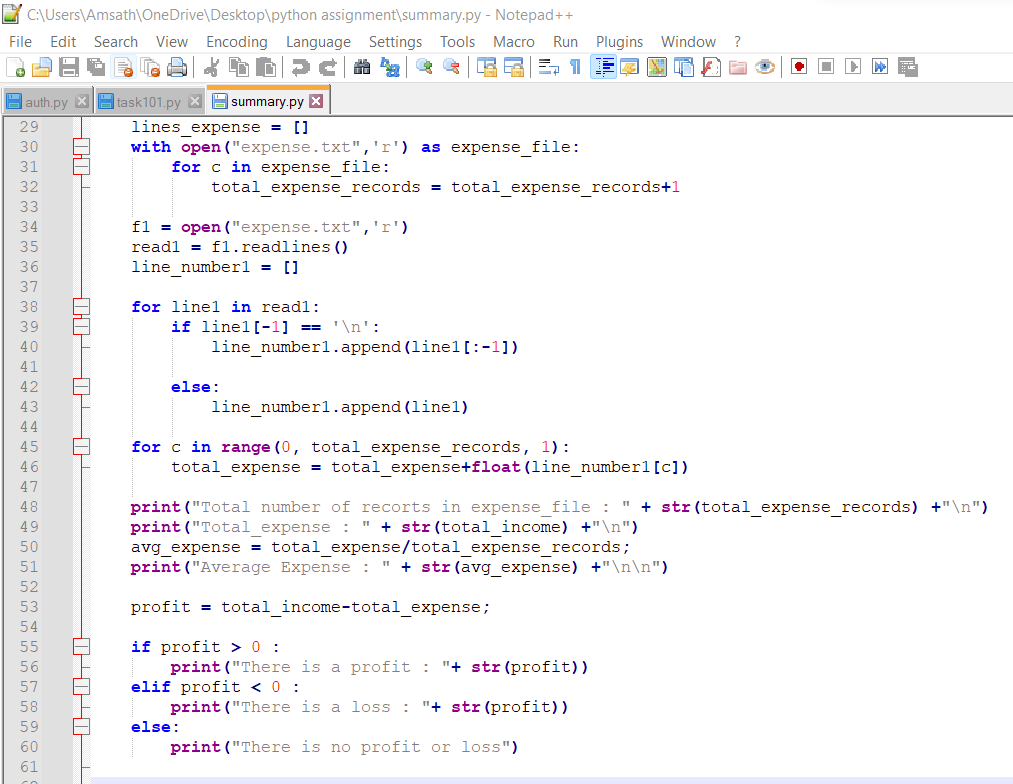
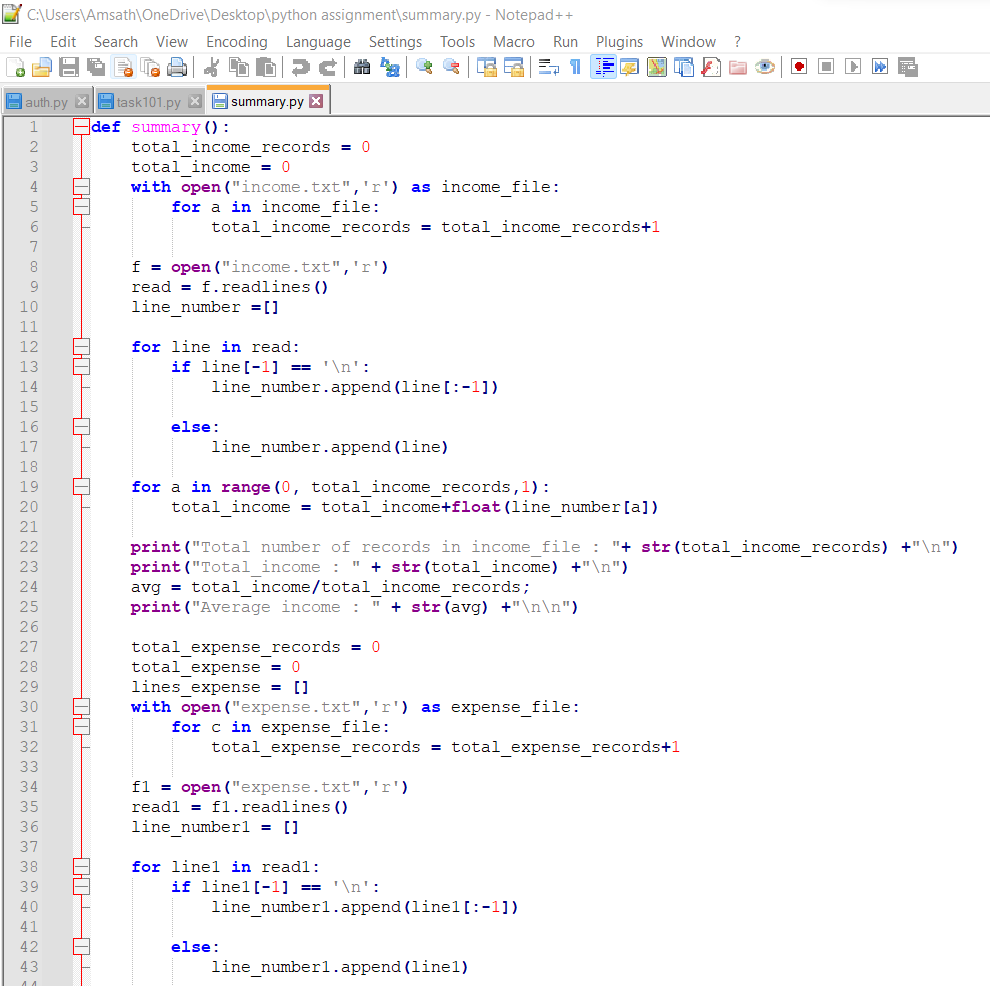
auth.py



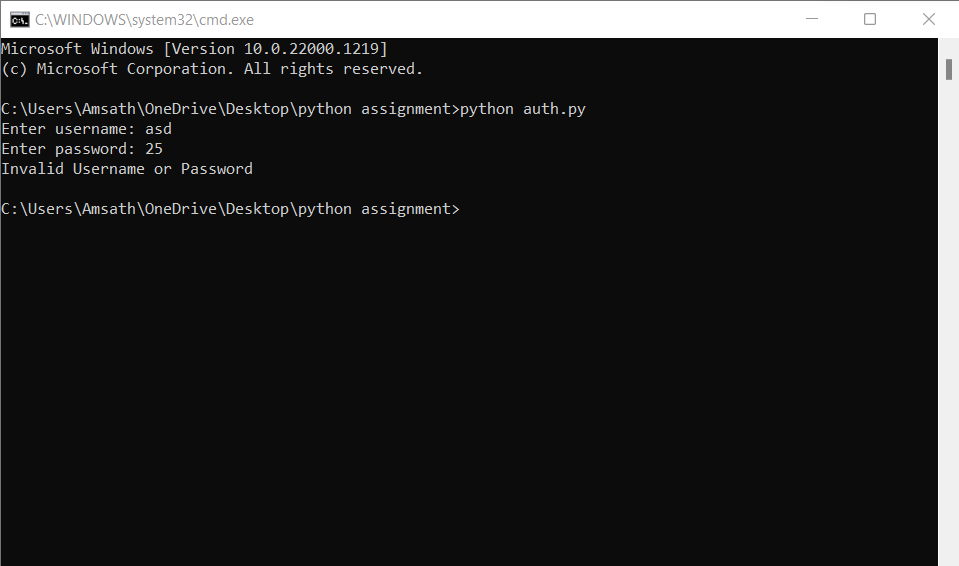
**task101.py**



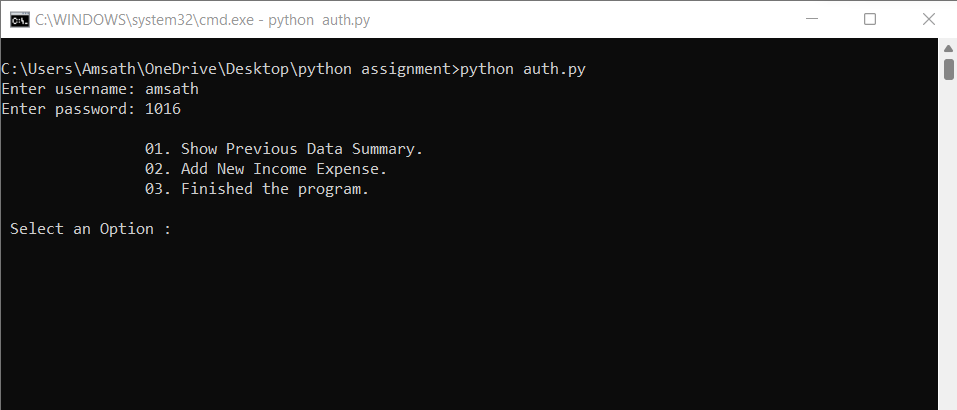
**Summary.py**



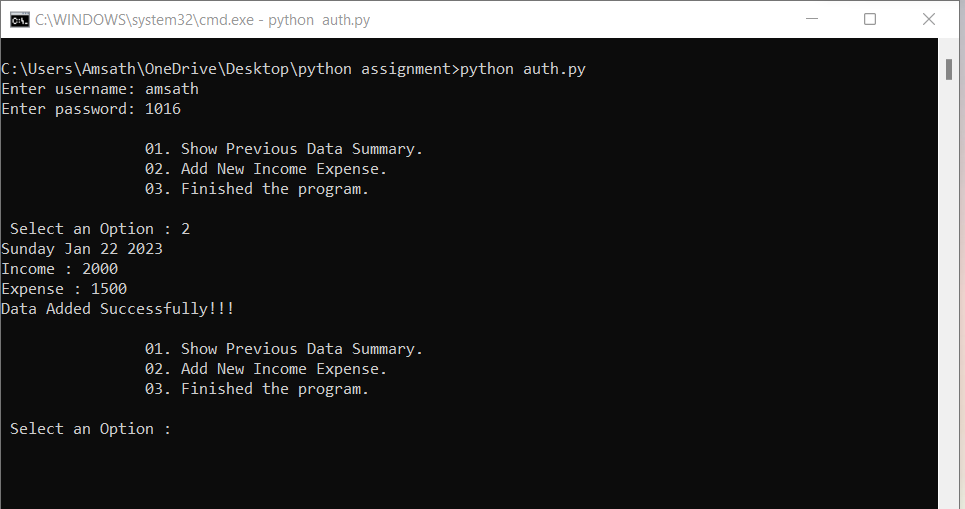
Invalid login



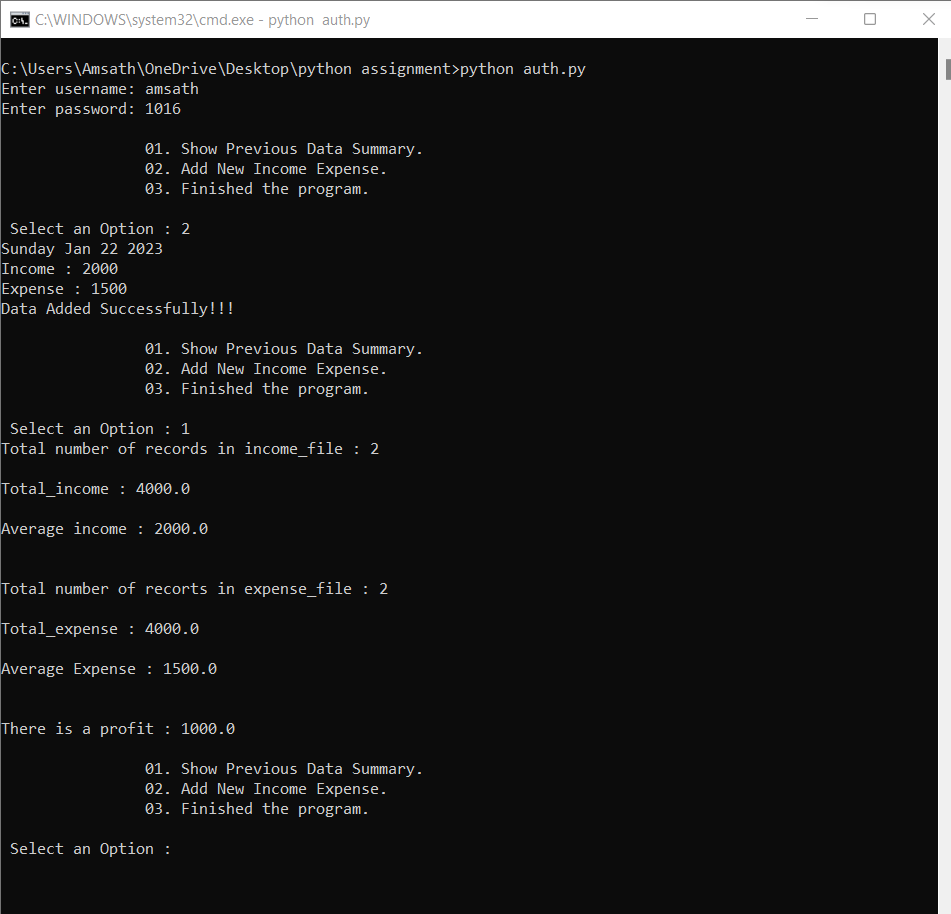
Successful login



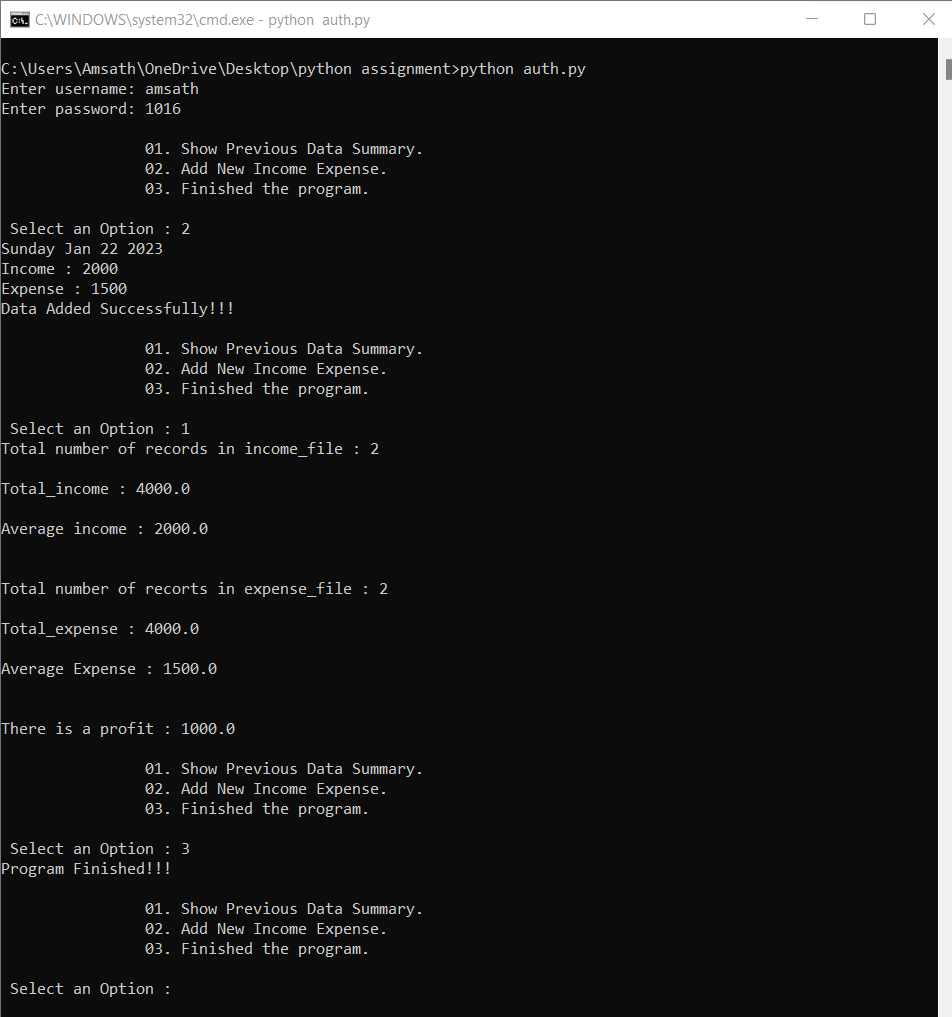
Add new data



Show all records



Finished the program



Task 4\s4.1 Explain the debugging procedure and the debugging tools provided by the IDE you used to construct the method in Task 3.

Starting a debugger session is essentially the same as running the application normally. We don't have to do anything specific to begin a debugger session because the debugger is already attached in the background. If PyCharm allows us to run our program, we can use the same parameters to debug it.

• Visual Debugging: Because pdb is intimidating and the concept is hard to grasp, several programmers still debug via print statements. The Python debugging GUI in PyCharm makes using a debugger straightforward by giving the process a visual face. Starting out is easy, and moving on to the essential debugging features is simple.

• Debugging TDD, or test-driven development, involves experimentation as the tests are being created. To help you with your exploration, set breakpoints in the context you're examining.

• Breakpoints: All debuggers have breakpoints, but only a few of them have breakpoints that are particularly adaptive. Have you ever kept pressing the "continue" button until you got to the loop iteration when your issue appeared? There's no need for it with PyCharm's conditional breakpoints. Sometimes all you need to do while code is running is check a variable's value. Breakpoints in PyCharm can be set up such that they simply report a message rather than pausing your program. Because exceptions can ruin your day, PyCharm's debugger can trip on them even if you are unsure of their origin.

• Variable Values at a Glance: PyCharm will display all of your variable values inline within your code when it encounters a breakpoint. To make it straightforward to notice which values have changed since you last reached the breakpoint, modified values are underlined.

• Watches: To customize your variable view, add watches to it. Whether they are simple or complex, you will be able to view exactly what you want to see.

4.2 You should include in your report

I. Describe the coding convention you followed in Task 3's code.

• Function length is short

• No global variables are utilized

• Indentation is used

• Variable names are clear and intelligible

• The GOTO statement is never used.

II. Consider how the debugging process can be utilized to create applications that are more robust, secure, and reliable.

It simply cannot. System defects are recreated, found, and described throughout the debugging process. It serves as a fix for problems that have already existed. Users in the field or testing are the two methods used to find these issues. This is what you do when a problem is acknowledged. Because of this, debugging is essential to improve application robustness even while it cannot help design more secure programs.

Since debugging is a reactive process, it is not a good way to boost security or robustness. Although testing is preferred, an application cannot be completely tested.

3. Analyze the reasons a coding standard is essential for both the team and the individual.

A coding standard makes sure that all developers involved in the project follow certain guidelines. The code is easy to read and maintains a sufficient level of consistency.

Consistency should be maintained when coding as it has a positive impact on program quality. Additionally, it should be made sure that all system layers adhere to the same coding standards and that they do not conflict with one another. The finished software code should be visible.

In terms of programming, think of the you that is right now and the you that is coming. Really. One factor is that you will always grow and learn, therefore for the most part, old code will smell terrible.

On the other side, as you investigate a problem, you'll learn a lot of recent, precise facts. A piece of that knowledge will be essentially lost as soon as the issue is resolved. In the future, you won't remember how you did it or what your strategy was. You'll also forget the rationale behind your choice of particular names and signatures.